Amendments to the Specification:

Please replace the paragraph on page 6, lines 8-20 with the following amended paragraph:

Referring to Figure 1, a network system 10 comprises a plurality of wireless devices 100 for interacting with one or more generic services 106, via a coupled Wide Area Network (WAN) 104 such as but not limited to the Internet. These devices 100 can be such as but not limited to, PDAs, pagers, cellular phones and the like. The generic services provided by the service 106 can be Web Services and/or other services such as but not limited to SQL Databases, IDL-based CORBA and RMI/IIOP systems, Legacy Databases, ~~J2EE~~ J2EE®, SAP RFCs, and COM/DCOM components. Further, the system 10 can also have a wireless network 102 for connecting the wireless devices 100 to the WAN 104. It is recognized that other devices (not shown) could be connected to the web service 106 via the WAN 104 and associated networks other than as shown in Figure 1. Web services 106 defined according to a schema are selected for the following description of the system 10, for the sake of simplicity. However, it is recognized that other services could be substituted for the web services 106, if desired. Further, the networks 102, 104 of the system 10 will hereafter be referred to as the network 104, for the sake of simplicity.

Please replace the paragraph on page 7, lines 14-31 with the following amended paragraph:

Referring to Figure 1, the web service 106 provides the information messages 105 which are used by the client application programs 302 (see Figure 2) on the devices 100 (either synchronously or asynchronously. Alternatively, or in addition, the web service 106 may receive and use the information messages 105 provided by the client application programs 302 executed on the devices 100, or perform tasks on behalf of client application programs 302 executed on the devices 100. The web service 106 can be defined as a software service, which can implement an interface such as expressed using Web Services Description Language (WSDL) registered in Universal Discovery Description and

Integration (UDDI) in a web services registry, and can communicate through messages 105 with client devices 100 by being exposed over the network 104 through an appropriate protocol such as the Simple Object Access Protocol (SOAP). In some implementations, SOAP is a specification that defines the XML format for the messages 105, including a well-formed XML fragment enclosed in SOAP elements. SOAP also supports document style applications where the SOAP message 105 is a wrapper around an XML document. A further optional part of SOAP defines the HTTP binding (i.e. header), whereas some SOAP implementations support ~~MSMQ~~ MSMQ®, ~~MQ Series~~ MQSeries®, SMTP, or TCP/IP transport protocols. Alternatively, the web service 106 may use other known communication protocols, message 105 formats, and the interface may be expressed in other web services languages than described above.

Please replace the paragraph on page 8, lines 17-29 with the following amended paragraph:

Referring again to Figure 1, the client runtime environment provided by the devices 100 can be configured to make the devices 100 operate as web clients of the web services 106. It is recognized that the client runtime environment can also make the devices 100 clients of any other generic schema-defined services over the network 104. The client runtime environment of the devices 100 is preferably capable of generating, hosting and executing the client application programs 302 (which include data 400 and screen 402 component definitions – see Figure 4 and description herein below) on the device 100. Further, specific functions of the client runtime environment can include such as but not limited to support for language, coordinating memory allocation, networking, management of data during I/O operations, coordinating graphics on an output device of the devices 100 and providing access to core object oriented classes and supporting files/libraries. Examples of the runtime environments implemented by the devices 100 can include such as but not limited to Common Language Runtime (CLR) by ~~Microsoft~~ Microsoft® and Java Runtime Environment (JRE) by Sun ~~Microsystems~~ Microsystems®.

Please replace the paragraph on page 13, lines 1-11 with the following amended paragraph:

It is recognised that either the screen component 402 or data component 400 definitions contain the mapping 800, which defines the relationship between the data object 800 and screen element 802 or the relationship between an individual data field (or group of data fields) of the data object 800 with screen element 802. It is recognised that the data object 800 may be passed to the user interface 202 as a parameter. In this case the data field values of the data object 800 mapped to the screen element ~~804~~ 802 would be extracted from the passed parameter. For example, an edit control (screen element 802) defined in a screen field definition of the screen component 402 could be mapped into a data field definition of the linked data component 400 (i.e. a one to one mapping 804) or a choice control (screen element 802) defined in a screen field definition of the screen component 402 could be mapped into a particular data field definition of a collection of data components 400 (i.e. a one to many mapping 804).

Please replace the paragraph on page 18, line 28 – page 19 line 10 with the following amended paragraph:

Further, the data components 400 can consist of a series of data field definitions written in such as but not limited to HTML, XHTML, XML and other structured definition languages, wherein the data objects 800 (see Figure [[9]]8) are data instances according to the data field definitions. It is noted that data object definitions provide definitions of the structure of data fields and their corresponding data types, such that data objects 800 are instances of a particular data object definition. Data Fields comprise the data object definitions and every data field has an associated data type. Further, Complex Data Fields contain a structure of sub-data fields. The definitions of the data objects 800 are included in the data component 400 schema of the application 302. These data definitions provide a data model for defining the data objects 800 as used by the application 302. Accordingly, the Data Component 400 is a set of one or more data field definitions grouped together to define the format of corresponding data field values when instantiated as the data object 800. The data component 400 definitions can have a primary or a composite key or optionally be defined without a key. The keys can be used with the mappings 804 to provide a unique link between a pair of the mapped data 400 and screen 402 components.

Please replace the paragraph on page 20, line 31 – page 21 line 7 with the following amended paragraph:

Referring again to Figure 4, the remaining message and other workflow parts ~~105~~ 405 of the application 302 can be provided by hard-coded application 302 elements and/or can be provided as further components 404, 406 described in the structured definition language and code/script respectively. In the case of using atomic message components 404, these define the format of messages 105 used by the component application program 302 to communicate with external systems such as the web service 106, and include message data represented in for example native code or XML. For example, one of the message components 404 may describe such as but not limited to a message for placing an order which includes the unique identifier for the order, the status of the order, and notes associated with the order. Message component 404 definitions written in the structured definition language can uniquely represent (and map to) WSDL messages, and can be generated dynamically at runtime. Web Service messages 105 are defined within the context of operation and there is defined correlations between the message components 404 in the component application program 302 definition. This correlation could be done using predefined message parameters and/or through separate workflow components 406, as further defined below.

Please replace the paragraph on page 23, lines 25-31 with the following amended paragraph:

The wireless applications 302 can have predefined sets of screen elements 802 and data objects 800, with the relationship (mapping ~~800~~ 804) between these sets being identified at application design phase in the field definitions of the data components 400 and the screen components 404. This model can be referred to as a predefined screen and data relationship model, such that the mapping 804 between the screen elements 802 presented to the user and data objects 800 can be expressed in metadata in the component field definitions using structured language such as XML.

Please replace the paragraph on page 25, lines 24-28 with the following amended paragraph:

Typically, the data object 800 of the data component 400 with values for the mapped screen elements 802 is passed to the screen of the user interface 202 as a parameter. In the Example screen component ~~404~~ 402 given above, the screen '**scrUserInfo**' received the data object 800 of a user as the parameter. The screen elements 802 on this screen mapped to the data field definition of the User data component 400 would display a data value from this parameter.